

Создать интерактивную игру в консоли, основанную на принципе "камень-ножницы-бумага". Оценка pylint - не меньше 8.

1. Требования проекта:

- для проекта должно быть создано виртуальное окружение.
- проект должен находиться в отдельной папке.
- для проекта должен быть создан гит репозиторий.
- файлы, которые не должны быть версионированы нужно добавить в .gitignore.
- ветка master репозитория должна содержать два пустых файла: requirements.txt и scores.txt.
- разработка должна вестись в ветке develop.
- задание будет приниматься исключительно в виде pull request из ветки develop в master

2. Базовая концепция:

- При запуске файла предложить пользователю ввести своё имя.
- Предложить пользователю ввести start для начала игры.
- После ввода start пользователь атакует - выбирает волшебника, воина или разбойника (1, 2, 3).
- Ход противника определяется автоматически.
- Волшебник побеждает воина. Воин побеждает разбойника. Разбойник побеждает волшебника.
- После атаки вывести результат раунда (Попал, промахнулся, ничья)
- Следующим атакует противник: пользователю предлагается выбрать защиту.
- При успешной атаке у противника уменьшается жизнь, игрок получает одно очко.
- При неудачной защите игрок теряет одну жизнь.
- Когда у игрока заканчиваются жизни - игра окончена.
- Когда у противника заканчиваются жизни игрок получает дополнительно пять очков.
- При завершении игры вывести результат игры.

3. Структура проекта:

- проект должен содержать файлы requirements.txt и scores.txt для хранения необходимых пакетов и сохранение таблицы рекордов соответственно.
- в файле settings.py должны храниться все константы.
- файл exceptions.py должен содержать кастомные исключения созданные для контроля игрового процесса и необходимый для них функционал.
- файл models.py должен содержать классы игрока и противника.
- файл game.py - основной исполняемый файл, в котором запускается игровой процесс.

4. exceptions.py:

- Содержит класс GameOver - унаследованный от Exception. В классе должен быть реализован механизм сохранения финального счета игрока по завершению игры в файл scores.txt

- Содержит класс EnemyDown - унаследованный от Exception. Функционал не обязателен.
 - Создать механизм сохранения только 10 лучших счетов игроков. Можно реализовать через класс Score.
5. models.py - class Enemy:
- свойства - level, lives.
 - конструктор принимает уровень. Уровень жизней противника = уровень противника.
 - содержит два метода:
 1. Статический - select_attack(): возвращает случайное число от одного до трёх.
 2. decrease_lives(self): уменьшает количество жизней. Когда жизней становится 0 вызывает исключение EnemyDown.
6. models.py - class Player:
- свойства: name, lives, score, allowed_attacks.
 - конструктор принимает имя игрока. Количество жизней указывается из настроек. Счет равен нулю.
 - методы: статический fight(attack, defense) - возвращает результат раунда - 0 если ничья, -1 если атака неуспешна, 1 если атака успешна.
 decrease_lives(self) - то же, что и Enemy.decrease_lives(), вызывает исключение GameOver.
 attack(self, enemy_obj) - получает ввод от пользователя (1, 2, 3), выбирает атаку противника из объекта enemy_obj; вызывает метод fight(); Если результат боя 0 - вывести "It's a draw!", если 1 = "You attacked successfully!" и уменьшает количество жизней противника на 1, если -1 = "You missed!"
 defence(self, enemy_obj) - то же самое, что и метод attack(), только в метод fight первым передается атака противника, и при удачной атаке противника вызывается метод decrease_lives игрока.
7. game.py:
- Содержит блок на проверку имени модуля (main)
 - внутри if блок try/except.
 - try запускает функцию play()
 - except обрабатывает два исключения: GameOver - выводит сообщение об ошибке, записывает результат в таблицу рекордов. KeyboardInterrupt - pass
 - finally печатает "Good bye!"
8. game.py - play():
- Ввод имени игрока
 - Создание объекта player
 - level = 1
 - Создание объекта enemy
 - в бесконечном цикле вызывает методы attack и defense объекта player
 - при возникновении исключения EnemyDown повышает уровень игры,

создает новый объект Enemy с новым уровнем, добавляет игроку +5 очков.

9. Дополнительные задания:

- Добавить валидацию пользовательского ввода.
- Использовать enum для хранения разрешенных атак.
- Расширить игровое меню:
 1. Добавить команду show scores - которая выводит записи из файла scores.txt
 2. Добавить команду exit - вызывает исключение и завершает работу программы.
 3. Добавить команду help - выводит список возможных команд (хранить в файле настроек).